3º CAPÍTULO

Análise de Imagens Binárias

Prof. Arnaldo Abrantes / anotado por Prof. Nuno Pinho da Silva

Índice

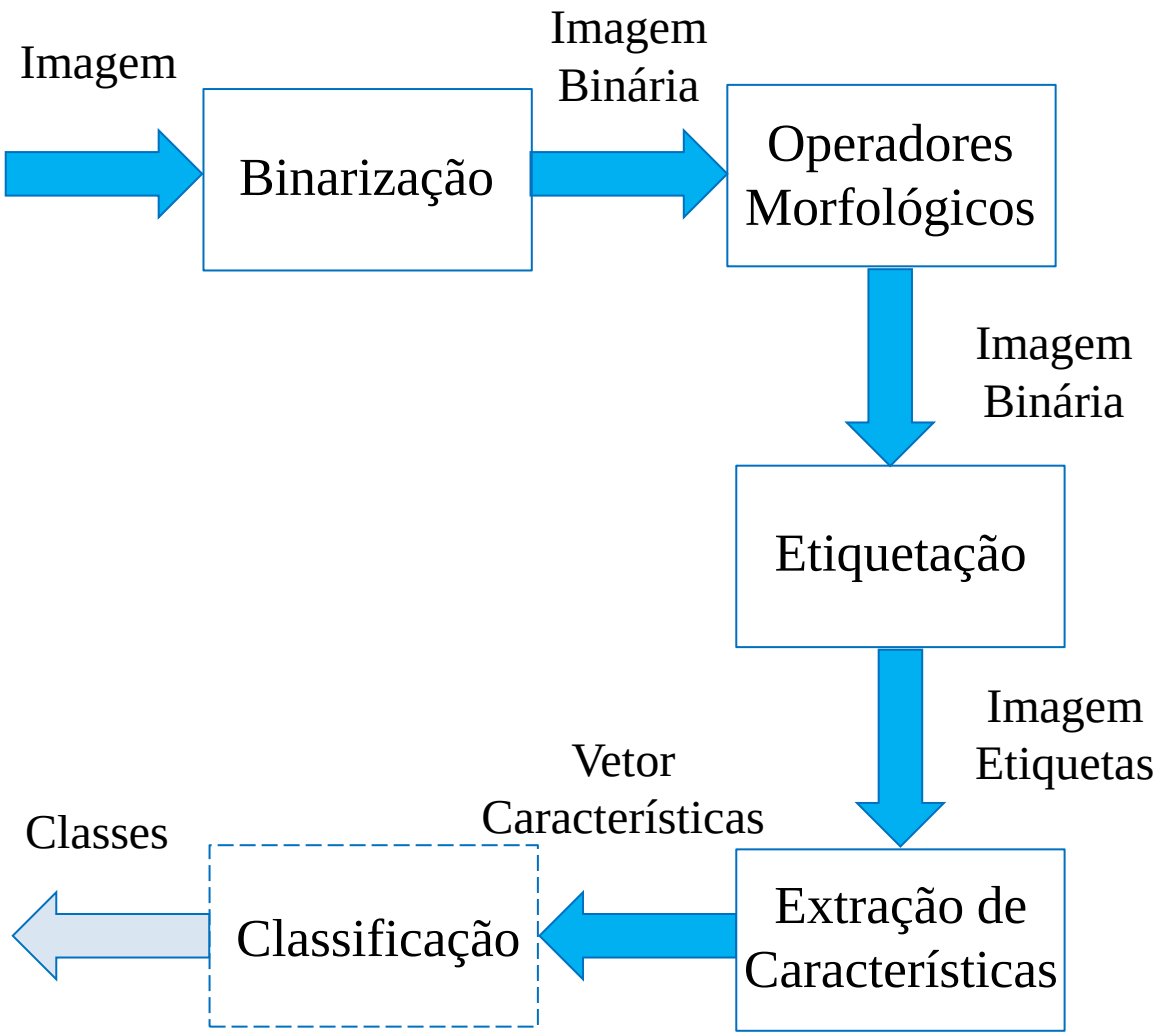


Imagem
Binarização
Imagem Binária
Operadores Morfológicos
Imagem Binária
Etiquetação
Imagem Etiquetas
Extração de Características
Vetor Características
Classificação
Classes

Pixels e vizinhanças

- Vizinhanças mais comuns

	N	
W	*	E
	S	

Vizinhança N_4

NW	N	NE
W	*	E
SW	S	SE

Vizinhança N_8

- Utilização de máscaras

Máscara: Conjunto de posições e respetivos pesos.

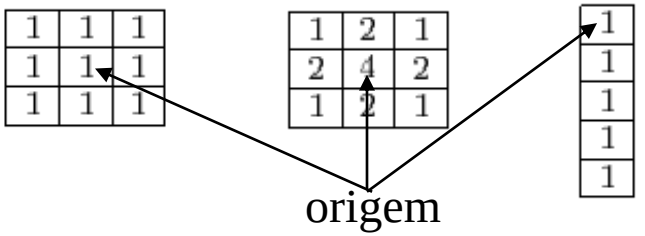
A aplicação da máscara a uma imagem gera uma imagem de saída com a mesma dimensão que a imagem original.

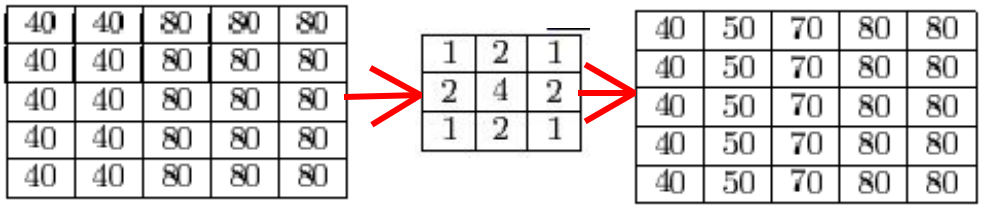
Para gerar um pixel na imagem de saída, a máscara é colocada sobre a imagem original com origem nesse pixel.

Os pesos da máscara são multiplicados pelos valores dos pixéis correspondentes e o resultado somado para gerar o valor do pixel de saída.

E para os pixéis na fronteira da imagem?

– Exemplo:



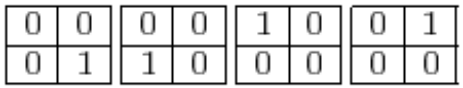


Resultado normalizado: Valores dos pixels divididos pelas some da máscara (=16)

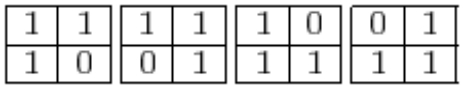
entrada → 

saída → 

- Algoritmo
  - ***Hipótese:*** Objecto é um conjunto conexo de pixels (conectividade 4) e sem buracos no seu interior



Cantos exteriores



Cantos interiores

```
Compute the number of foreground objects of binary image B.
Objects are 4-connected and simply connected.
E is the number of external corners.
I is the number of internal corners.

    procedure count_objects(B);
    {
    E := 0;
    I := 0;
    for L := 0 to MaxRow - 1
       for P := 0 to MaxCol - 1
          {
          if external_match(L, P) then E := E + 1;
          if internal_match(L, P) then I := I + 1;
          };
    return((E - I) / 4);
    }
```

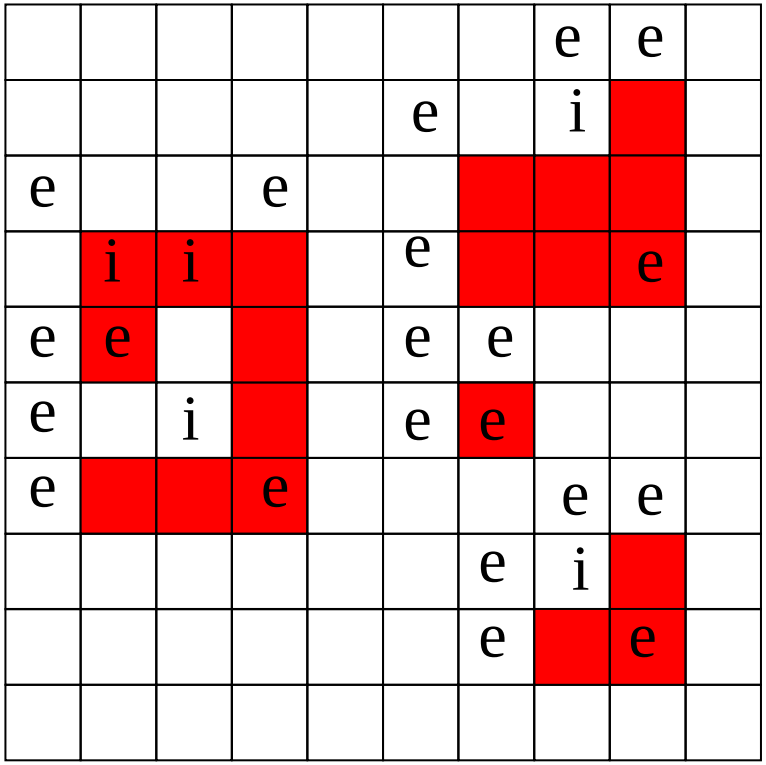
Outra Aplicação de Máscaras: Encontrar padrões em imagens: retorna 'True' caso o padrão se encontre na imagem.

A função **external_match** testa as máscaras "cantos exteriores" e devolve true caso o padrão da imagem correspondente à aplicação das máscaras seja um dos "cantos exteriores".

A função **internal_match** testa as máscaras "cantos interiores" e devolve true caso o padrão da imagem correspondente à aplicação das máscaras seja um dos "cantos interiores".

Porque é que se divide por 4?

Quantos objectos?

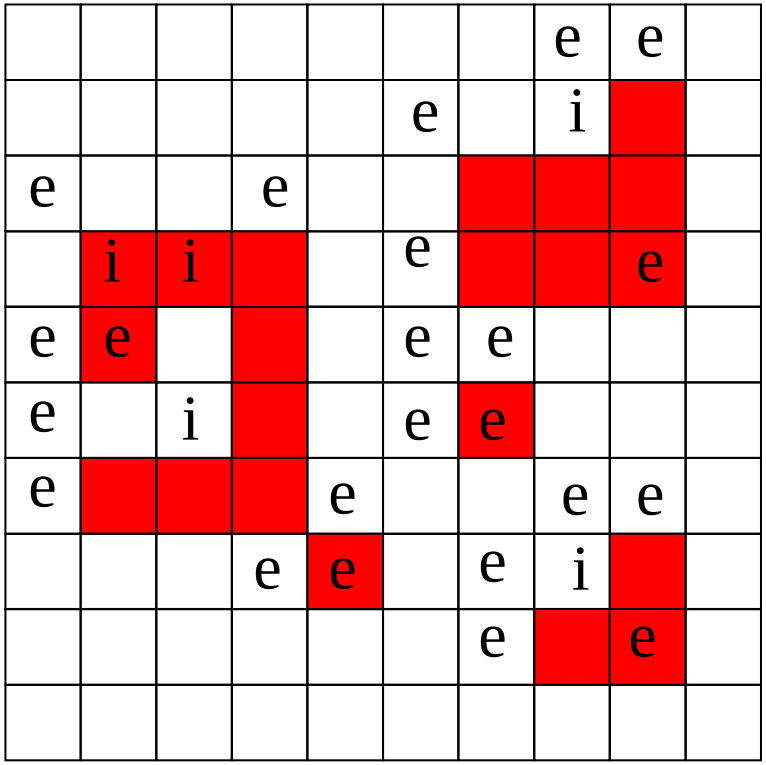


e - 21

i - 5

$$\#= \frac{21-5}{4} = \frac{16}{4} = 4$$

E agora?



e - 23

i - 4

$$\#= \frac{23-4}{4} = \frac{19}{4} = ?$$

ECC - algoritmo clássico

1. Scan the binary image left to right, top to bottom.

Linha a linha
(row-by-row labeling algorithm)

2. If an unlabeled pixel has a value of '1', assign a new label to it according to the following rules:

$$\begin{matrix} & 0 \\ 0 & 1 \end{matrix} \rightarrow \begin{matrix} & 0 \\ 0 & \mathrm{L} \end{matrix} \qquad \begin{matrix} & 0 \\ \mathrm{L} & 1 \end{matrix} \rightarrow \begin{matrix} & 0 \\ \mathrm{L} & \mathrm{L} \end{matrix}$$

$$\begin{matrix} & \mathrm{L} \\ 0 & 1 \end{matrix} \rightarrow \begin{matrix} & \mathrm{L} \\ 0 & \mathrm{L} \end{matrix} \qquad \begin{matrix} & \mathrm{L} \\ \mathrm{M} & 1 \end{matrix} \rightarrow \begin{matrix} & \mathrm{L} \\ \mathrm{M} & \mathrm{L} \end{matrix} \quad (\text{Set } L = M).$$

3. Determine equivalence classes of labels.

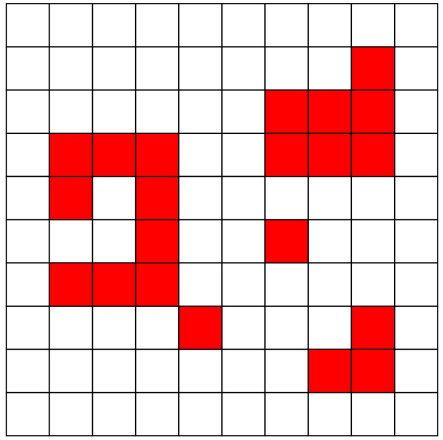
4. In the second pass, assign the same label to all elements in an equivalence class.

Figure 3.8: Sequential Connected Component Algorithm.

Assuma que no final da primeira passagem encontrou etiquetas {a,....,l} e os seguintes pares de etiquetas iguais
(a=b), (l=k), (c=f), (a=g), (b=e), (j=l), (h=f), e (i=k).

Assim, temos as seguintes classes equivalentes
(a=b=e=g), (c=f=h), (i=j=l=k), (d)

Aplique o ECC clássico à imagem



Altere o algoritmo para reconhecer componentes com conectividade-8. Quantos objetos tem a imagem?

Extracção de componentes conexos - algoritmo recursivo

Compute the connected components of a binary image.
B is the original binary image.
LB will be the labeled connected component image.

Assume que a imagem pode ser completamente carregada na memória

```
procedure recursive_connected_components(B, LB);
{
LB := negate(B);
label := 0;
find_components(LB, label);
print(LB);
}

procedure find_components(LB, label);
{
for L := 0 to MaxRow
   for P := 0 to MaxCol
      if LB[L,P] == -1 then
         {
         label := label + 1;
         search(LB, label, L, P);
         }
}

procedure search(LB, label, L, P);
{
LB[L,P] := label;
Nset := neighbors(L, P);
for each (L',P') in Nset
   {
   if LB[L',P'] == -1
   then search(LB, label, L', P');
   }
}
```

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1

entrada

saída

1	1	0	1	1	1	0	2
1	1	0	1	0	1	0	2
1	1	1	1	0	0	0	2
0	0	0	0	0	0	0	2
3	3	3	3	0	4	0	2
0	0	0	3	0	4	0	2
5	5	0	3	0	0	0	2
5	5	0	3	0	2	2	2

Algoritmo recursivo - Exemplo

Step 1.

-1	-1	0	-1	-1	-1
-1	-1	0	-1	0	0
-1	-1	-1	-1	0	0

Step 2.

1	**-1**	0	-1	-1	-1
-1	-1	0	-1	0	0
-1	-1	-1	-1	0	0

Step 3.

1	1	0	-1	-1	-1
-1	-1	0	-1	0	0
-1	-1	-1	-1	0	0

Step 4.

1	1	0	-1	-1	-1
1	**-1**	0	-1	0	0
-1	-1	-1	-1	0	0

Step 5.

1	1	0	-1	-1	-1
1	1	0	-1	0	0
-1	-1	-1	-1	0	0

neighbors(L,P) recebe as coordenadas espaciais de um píxel (L,P). Retorna o conjunto de coordenadas de todos os píxeis vizinhos, utilizando vizinhança-4 ou vizinhança-8. Retorna os viznhos conforme a ordem das Figuras.

Somente coordenadas admissíveis são retornadas

Vizinhança 4

	1	
2	*	3
	4	

Vizinhança 8

1	2	3
4	*	5
6	7	8

Se (L,P)=(MaxRow-1,MaxCol) e considerarmos uma vizinhança-8, quais as coordenadas retornadas por neibors(L,P)?

Estrutura União-Procura

Construct the union of two sets.
X is the label of the first set.
Y is the label of the second set.
PARENT is the array containing the union-find data structure.

```
procedure union(X, Y, PARENT);
{
j := X;
k := Y;
while PARENT[j] <> 0
   j := PARENT[j];
while PARENT[k] <> 0
   k := PARENT[k];
if j <> k then PARENT[k] := j;
}
```

union(X,Y,PARENT) recebe duas etiquetas (labels) e o array PARENT. Modifica a estrutura (se necessário) para juntar o conjunto que contém X com o conjunto que contém Y.

Começa ns etiquetas X e Y e segue os array PARENT até atingir as raízes das árvores que contêm X e Y.

Se as raízes não forem iguais, uma etiqueta (label) é feita pai da outra, e.g. no exemplo, X é pai de Y, arbitrariamente). Pode calcular-se o tamanho das árvores e associar a árvore mais pequena à raiz da árvore maior.

Find the parent label of a set.
X is a label of the set.
PARENT is the array containing the union-find data structure.

```
procedure find(X, PARENT);
{
j := X;
while PARENT[j] <> 0
   j := PARENT[j];
return(j);
}
```

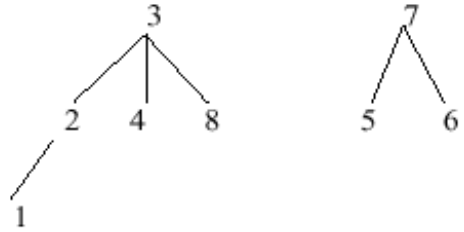
Recebe uma etiqueta (label) X e o array PARENT. Encontra a etiqueta da raiz que contém a etiqueta (label) X.

Cada conjunto é armazenado numa árvore, em que cada nó representa uma etiqueta e aponta para o seu nó pai.

Isto é realizado através do array PARENT cujos **índices** são o conjunto de possíveis etiquetas e **o valor em cada índice** é a etiqueta do nó pai.

PARENT

1	2	3	4	5	6	7	8
2	3	0	3	7	7	0	3



ECC - algoritmo clássico com união-procura

```
Compute the connected components of a binary imag
B is the original binary image.
LB will be the labeled connected component image.

    procedure classical_with_union-find(B,LB);
    {
    "Initialize structures."
    initialize();
    "Pass 1 assigns initial labels to each row L of the image."
    for L := 0 to MaxRow
      {
      "Initialize all labels on line L to zero"
      for P := 0 to MaxCol
        LB[L,P] := 0;
      "Process line L."
      for P := 0 to MaxCol
        if B[L,P] == 1 then
          {
          A := prior_neighbors(L,P);
          if isempty(A)
          then { M := label; label := label + 1; }
          else M := min(labels(A));
          LB[L,P] := M;
          for X in labels(A) and X <> M
            union(M, X, PARENT);
          }
      }
    "Pass 2 replaces Pass 1 labels with equivalence class labels."
    for L := 0 to MaxRow
      for P := 0 to MaxCol
        if B[L,P] == 1
        then LB[L,P] := find(LB[L,P],PARENT);
    } ;
```

Passo 1: Propaga a etiqueta (label) de um pixel para a direita e para baixo. Quando diferentes labels podem propagar para o mesmo pixel, a etiqueta menor é propagada e cada equivalência destas que é encontrada é introduzida na estrutura union-find. No final do Passo 1, as equivalências estão determinadas e com uma única etiqueta, correspondente à label da raiz da árvora na estrutura union-find.
Passo 2: Realiza a tradução das labels, atribuindo a cada pixel a etiqueta da sua classe de equivalência.

prior_neighbors: Retorna os vizinhos 1 pixel em cima e à esquerda do pixel dado. Pode ser codificada para vizinhança-4 (retorna os vizinhos a norte e oeste) ou para vizinhança-8 (retorna os vizinhos noroeste, norte, nordeste e oeste) utilizando uma máscara apropriada.
labels: retorna o conjunto atual de etiquetas (labels) de um dado conjunto de píxeis.

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1

← entrada

1º passo →

1	1	0	2	2	2	0	3
1	1	0	2	0	2	0	3
1	1	1	1	0	0	0	3
0	0	0	0	0	0	0	3
4	4	4	4	0	5	0	3
0	0	0	4	0	5	0	3
6	6	0	4	0	0	0	3
6	6	0	4	0	7	7	3

PARENT

1	**2**	**3**	**4**	**5**	**6**	**7**
0	1	0	0	0	0	3

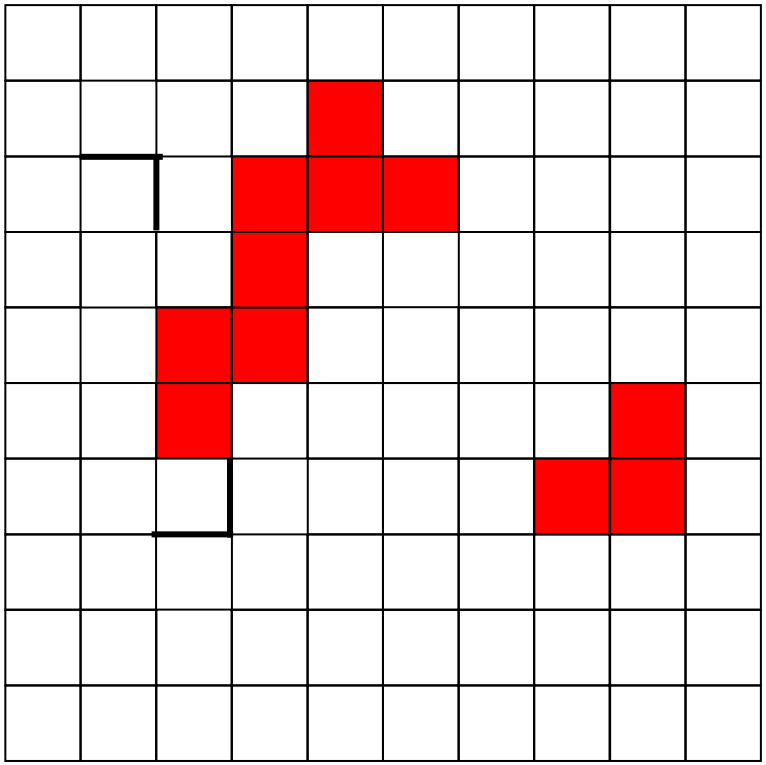
← classes equiv.

2º passo →

1	1	0	1	1	1	0	3
1	1	0	1	0	1	0	3
1	1	1	1	0	0	0	3
0	0	0	0	0	0	0	3
4	4	4	4	0	5	0	3
0	0	0	4	0	5	0	3
6	6	0	4	0	0	0	3
6	6	0	4	0	3	3	3

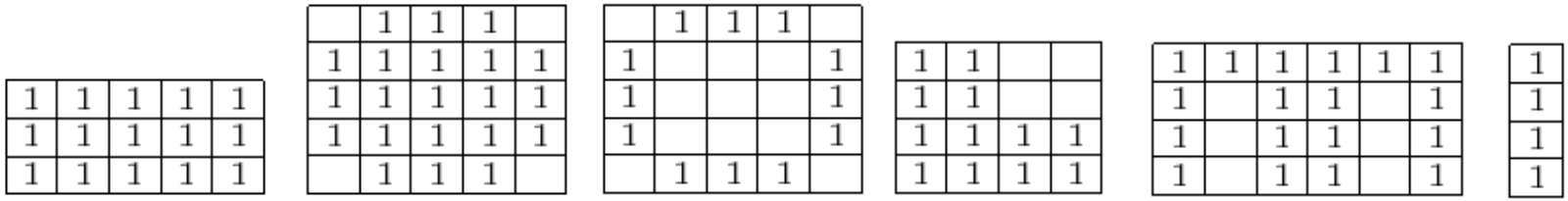
Exercício

- Aplique o algoritmo de ECC, considerando conectividade 4



- Qual deveria ser a forma da máscara se considerasse conectividade 8?

- Elementos estruturantes



ones(3,5) disco(5) anel(5)

  - necessário definir uma origem

- ***Definição***: A **dilatação** duma imagem binária *B* pelo elemento estruturante *S* define-se da seguinte forma

$$B \oplus S = \bigcup_{b \in B} S_b \qquad S_b = \{s + b | s \in S\}$$

- ***Definição:*** A **erosão** duma imagem binária *B* pelo elemento estruturante *S* define-se da seguinte forma

$$B \ominus S = \{b | b + s \in B \forall s \in S\}$$

1	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

a) Binary image B

1	1	1
1	**1**	1
1	1	1

b) Structuring Element S

1. Calcule a imagem resultante da B$\oplus$S
2. Calcule a imagem resultante da B$\ominus$S

A imagem de saída tem a mesma dimensão que a imagem de entrada e é inicializada com todos os píxeis a '0'.

Operadores morfológicos - Exemplos

1	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

a) Binary image B

1	1	1
1	**1**	1
1	1	1

b) Structuring Element S

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1			

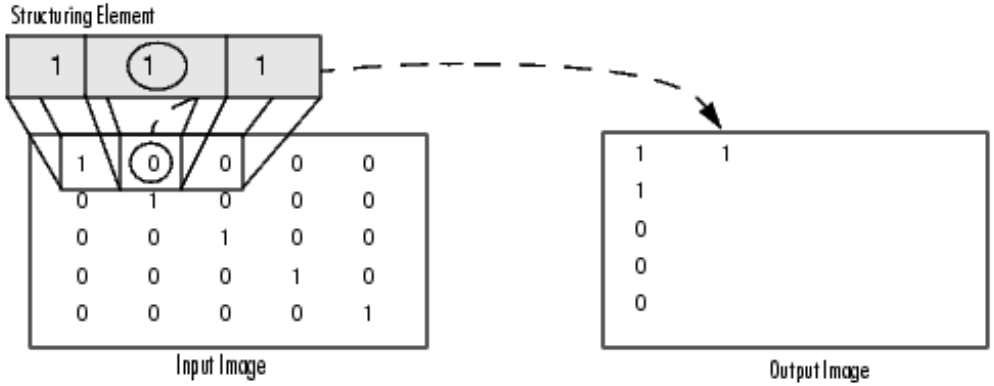
c) Dilation $B \oplus S$

				1	1		
				1	1		
				1	1		

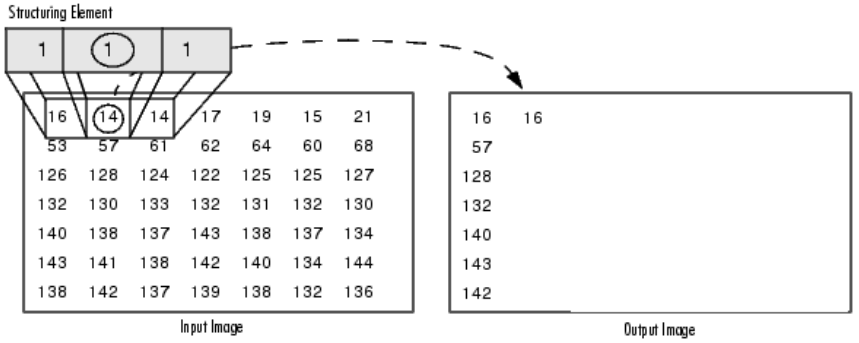
d) Erosion $B \ominus S$

Dilatação e erosão – Generalização para imagens monocromáticas

Operação	Regra
Dilatação	O valor do pixel de saída é o valor **máximo** de todos os pixeis na vizinhança do pixel de entrada. É atribuído o valor mínimo (0) aos pixeis exteriores
Erosão	O valor do pixel de saída é o valor **mínimo** de todos os pixeis na vizinhança do pixel de entrada. É atribuído o valor máximo (1 ou 255) aos pixeis exteriores



Dilatação de imagem binária



Dilatação de imagem monocromática

- ***Definição:*** O **fecho** duma imagem binária *B* pelo elemento estruturante *S* define-se da seguinte forma

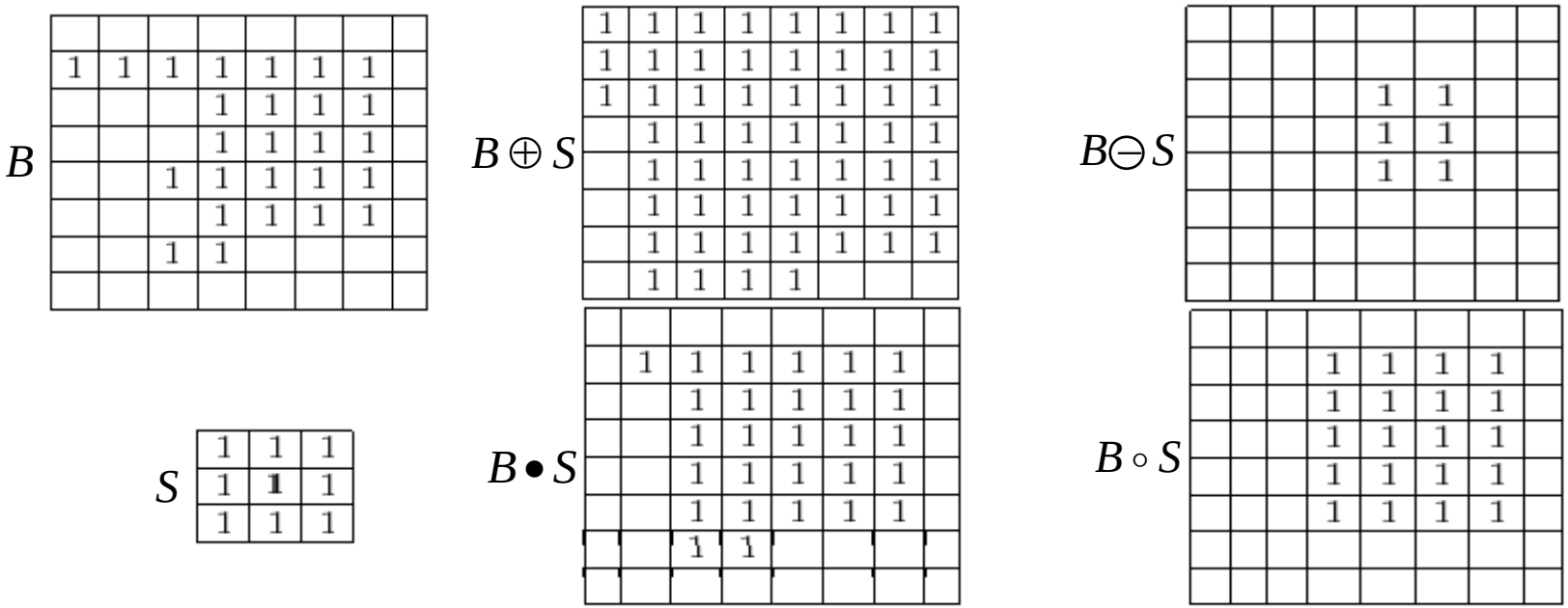
$$B \bullet S = (B \oplus S) \ominus S$$

Fecha buracos internos e elimina baías na região

- ***Definição:*** A **abertura** duma imagem binária *B* pelo elemento estruturante *S* define-se da seguinte forma

$$B \circ S = (B \ominus S) \oplus S$$

Elimina "pontões" na imagem



Exemplos de aplicação

- aplicações médicas (resolução 512x512)
  - **abertura** com disco(13) seguido de **fecho** com disco(2)

original

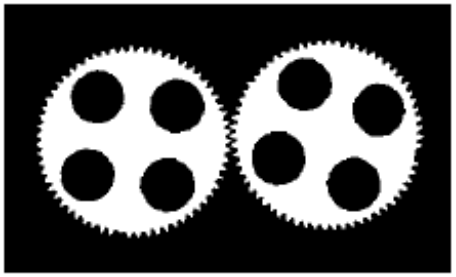
binarizada

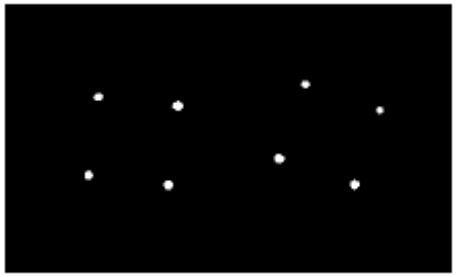
processada

- extracção de primitivas geométricas
  - subtrai da imagem original a obtida desta através do operador **abertura** usando pequeno disco como elemento estruturante

Inspecção

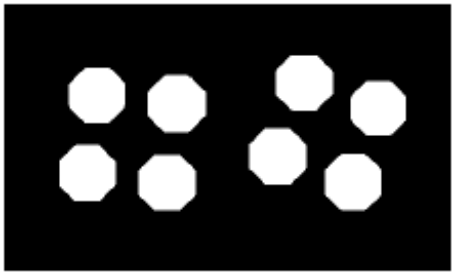
entrada →

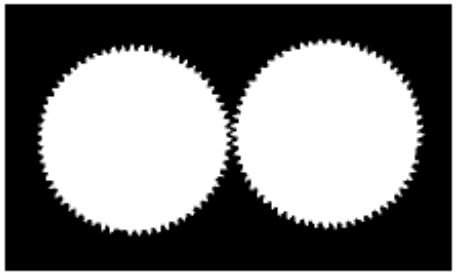




(2)=Erosão de (1) com um anel que contém tangencialmente os burados das rodas dentadas

(3)=Dilatação (2) com um octógono que contém tangencialmente os buracos das rodas dentadas





(4)=(3) OR (1)

(5)= Anel para selecionar os dentes

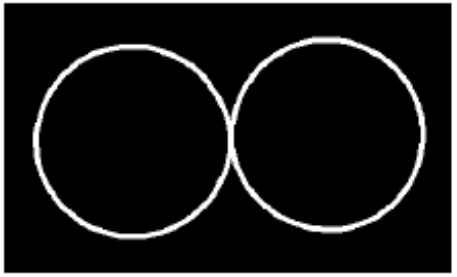
Realização:

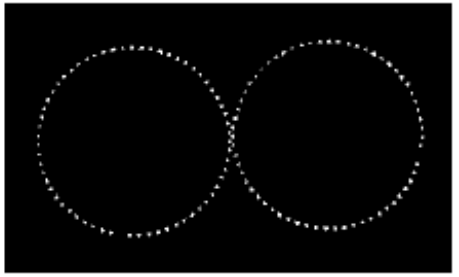
5.1. Aplica-se a (4) abertura com um disco que cobre as rodas dentadas menos os dentes, para eliminar os dentes,

5.2. Seguido de dilatação com um disco para ficarmos com uma àrea até à base dos dentes,

5.3. seguido de uma dilatação com um disco para ficarmos com uma área até à ponta dos dentes,

5.4. seguido de 5.3 - 5.2





(6)=(1) AND (5)

(7)=Dilatação com um disco cujo diâmetro liga as ponta de um par de dentes

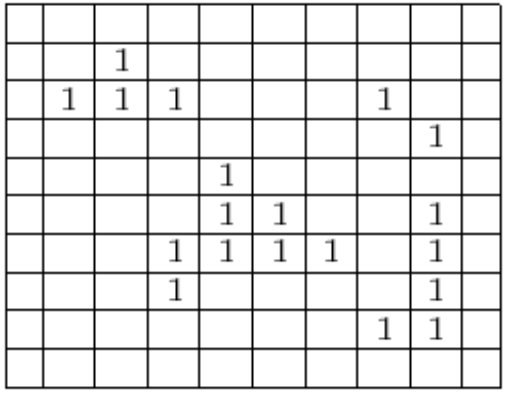
(8)= (5)-(7) seguido de dilatação com um dico para salientar os defeitos e OR com (7)

→ saída

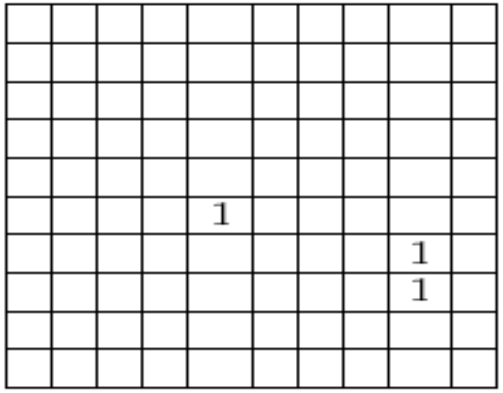
- ***Definição:*** Dadas as imagens binárias original *B*, e processada *C*, e o elemento estruturante *S*, e seja $C_0 = C$ e $C_n = (C_{n-1} \oplus S) \cap B$. A **dilatação condicional** de *C* por *S* com respeito a *B* define-se como $C \oplus \big|_B S = C_m$

  onde *m* é o menor inteiro que satisfaz a condição $C_m = C_{m-1}$

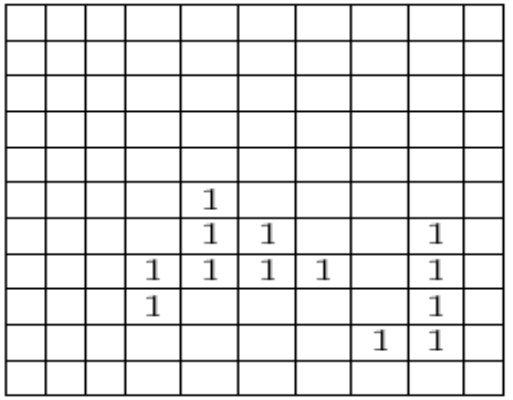
B



$C = B \ominus V$



$C \oplus \big|_B S$



C_{n-1} é repetidamente dilatado por S e reduzido aos valores '1' da imagem B, até o resultado não alterar.

V

1	
1	
1	

S

1	1	1
1	**1**	1
1	1	1

Propriedades de regiões

- Área

$$A = \sum_{(r,c)\in R} 1$$

- Centróide

$$\bar{r} = \frac{1}{A} \sum_{(r,c)\in R} r \qquad \bar{c} = \frac{1}{A} \sum_{(r,c)\in R} c$$

- Pixels de perímetro

$$P_4 = \{(r,c) \in R \,|\, N_8(r,c) - R \neq \varnothing\}$$

conectividade-4

$$P_8 = \{(r,c) \in R \,|\, N_4(r,c) - R \neq \varnothing\}$$

conectividade-8

  - comprimento do perímetro

$$|P| = \left|\{k \,|\, (r_{k+1}, c_{k+1}) \in N_4(r_k, c_k)\}\right| + \sqrt{2}\left|\{k \,|\, (r_{k+1}, c_{k+1}) \in N_8(r_k, c_k) - N_4(r_k, c_k)\}\right|$$

- Circularidade (1)

$$C_1 = \frac{|P|^2}{A}$$

Não é mínima para discos.

- Circularidade (2)

$$C_2 = \frac{\mu_R}{\sigma_R}$$

  - distância radial média

$$\mu_R = \frac{1}{K}\sum_{k=0}^{K-1}\left\|(r_k, c_k) - (\bar{r}, \bar{c})\right\|$$

  - desvio padrão da distância radial

$$\sigma_R = \left(\frac{1}{K}\sum_{k=0}^{K-1}\left(\left\|(r_k, c_k) - (\bar{r}, \bar{c})\right\| - \mu_R\right)^2\right)^{1/2}$$

A medida de circularidade C2 cresce monotonicamente à medida que a região se aproxima de um circulo e é igual para formas discretas (i.e. representadas por píxeis numa imagem) ou contínuas

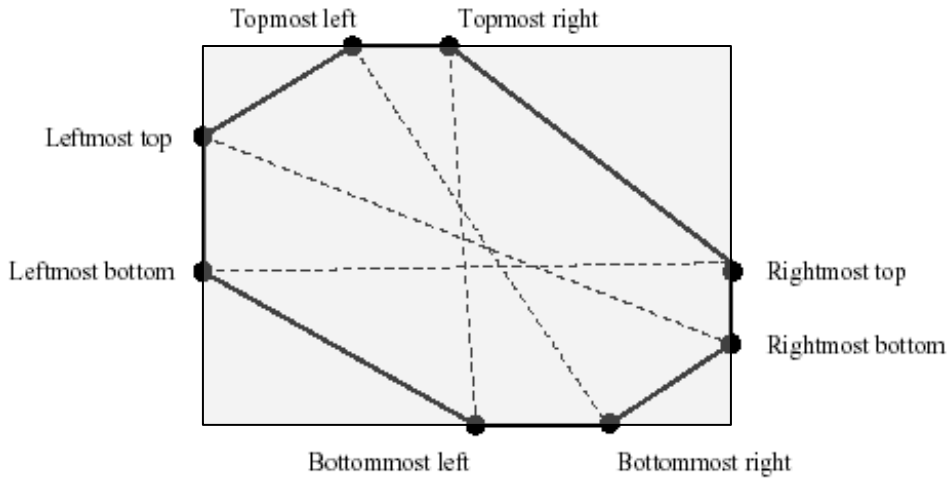
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
2	2	2	2	0	0	0	0	0	1	1	1	1	1	1	0
2	2	2	2	0	0	0	0	1	1	1	1	1	1	1	1
2	2	2	2	0	0	0	0	1	1	1	1	1	1	1	1
2	2	2	2	0	0	0	0	1	1	1	1	1	1	1	1
2	2	2	2	0	0	0	0	0	1	1	1	1	1	1	0
2	2	2	2	0	0	0	0	0	0	1	1	1	1	0	0
2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
2	2	2	2	0	0	3	3	3	0	0	0	0	0	0	0
2	2	2	2	0	0	3	3	3	0	0	0	0	0	0	0
2	2	2	2	0	0	3	3	3	0	0	0	0	0	0	0
2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0

Os píxeis k=0,...,K-1 pertencem ao perímetro P da região

region num.	region area	row of center	col of center	perim. length	circu-larity$_1$	circu-larity$_2$	radius mean	radius var.
1	44	6	11.5	21.2	10.2	15.4	3.33	.05
2	48	9	1.5	28	16.3	2.5	3.80	2.28
3	9	13	7	8	7.1	5.8	1.2	0.04

Propriedades – fronteiras e comprimentos

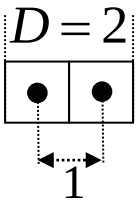
- Rectângulo (e octógono) de fronteira



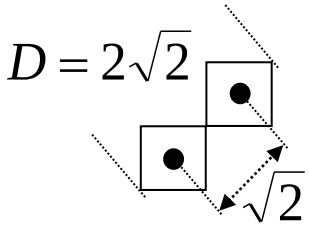
- Comprimento de um segmento (eixo)

$$D = \sqrt{(r_2 - r_1)^2 + (c_2 - c_1)^2} + Q(\theta)$$

$$Q(\theta) = \begin{cases} \frac{1}{|\cos(\theta)|} & : \quad |\theta| < 45^o \\ \frac{1}{|\sin(\theta)|} & : \quad |\theta| > 45^o \end{cases}$$



$\theta = 0^o$



$\theta = 45^o$

Propriedades – Momentos de 2ªordem

- Momentos de 2ªordem centrados

$$\mu_{rr} = \frac{1}{A}\sum_{(r,c)\in R}(r-\bar{r})^2 \qquad \mu_{cc} = \frac{1}{A}\sum_{(r,c)\in R}(c-\bar{c})^2 \qquad \mu_{rc} = \frac{1}{A}\sum_{(r,c)\in R}(r-\bar{r})(c-\bar{c})$$

- Relação entre momentos e regiões elípticas

$$R = \{(r,c) \mid dr^2 + 2erc + fc^2 \le 1\} \qquad \begin{pmatrix} d & e \\ e & f \end{pmatrix} = \frac{1}{4(\mu_{rr}\mu_{cc} - \mu_{rc}^2)}\begin{pmatrix} \mu_{cc} & -\mu_{rc} \\ -\mu_{rc} & \mu_{rr} \end{pmatrix}$$

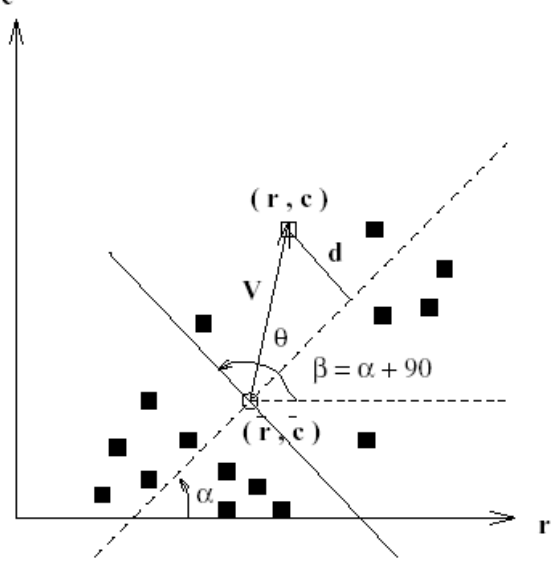
- Eixos de menor e maior inércia
  - Formulação

$$\mu_{\bar{r},\bar{c},\alpha} = \frac{1}{A}\sum_{(r,c)\in R} d^2$$

$$= \frac{1}{A}\sum_{(r,c)\in R}(\bar{V}\circ(\cos\beta,\sin\beta))^2$$

  - Solução

$$\tan(2\hat{\alpha}) = \frac{2\mu_{rc}}{\mu_{rr}-\mu_{cc}}$$



Grafos de adjacências de regiões

- ***Problema:*** regiões possuem buracos (fundo) no seu interior
- ***Solução:*** algoritmo com 3 passos
  - aplicação do algoritmo de extracção de componentes conexos duas vezes: (1) aos pixels activos e (2) aos pixels do fundo

Uma região é adjacente a outra se existirem pixéis de ambas as regiões que sejam vizinhos.

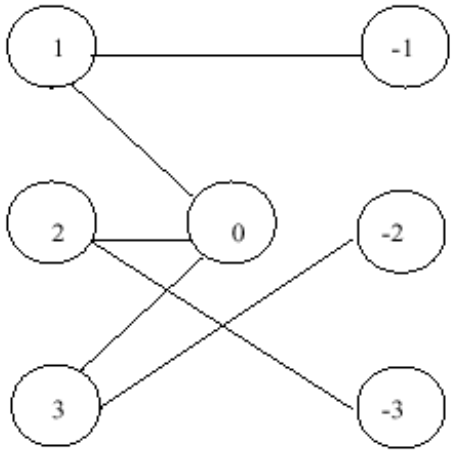
Se o fundo for uma única região conexa, não existe mais nada a calcular

0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	2	2	0
0	1	-1	-1	-1	1	0	2	2	0
0	1	1	1	1	1	0	2	2	0
0	0	0	0	0	0	0	2	2	0
0	3	3	3	0	2	2	2	2	0
0	3	-2	3	0	2	-3	-3	2	0
0	3	-2	3	0	2	-3	-3	2	0
0	3	3	3	0	2	2	2	2	0
0	0	0	0	0	0	0	0	0	0

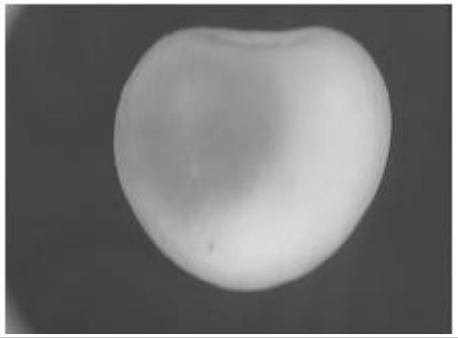
Buracos: região de background que é adjacente a apenas 1 região de foreground

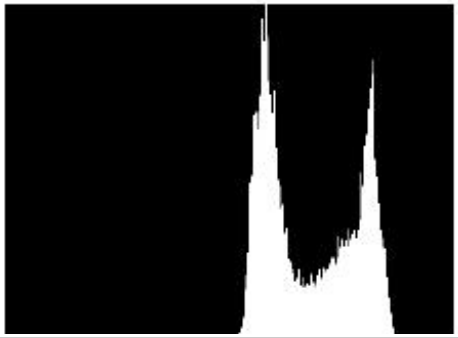
- (3) construção de grafo de relações

Um RAG é um grafo onde cada nó corresponde a uma região da imagem e um segmento liga duas regiões adjacentes.



- ***Definição:*** o histograma *h* da imagem monocromática *I* é definido por $h(m) = \left|\{(r,c) | I(r,c) = m\}\right|$

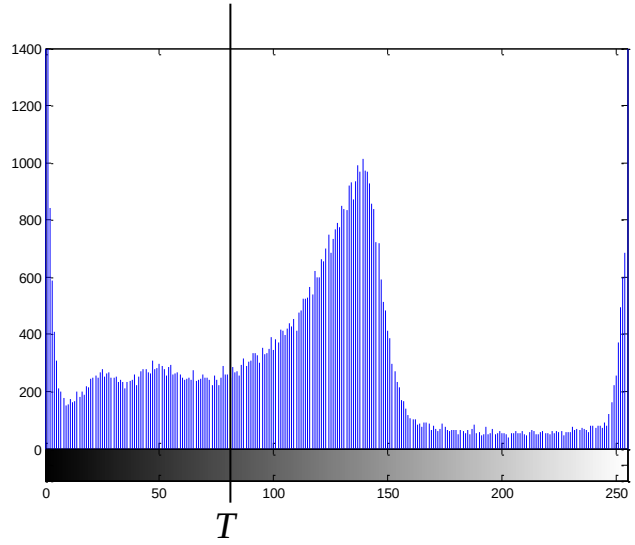




Compute the histogram H of gray-tone image I.

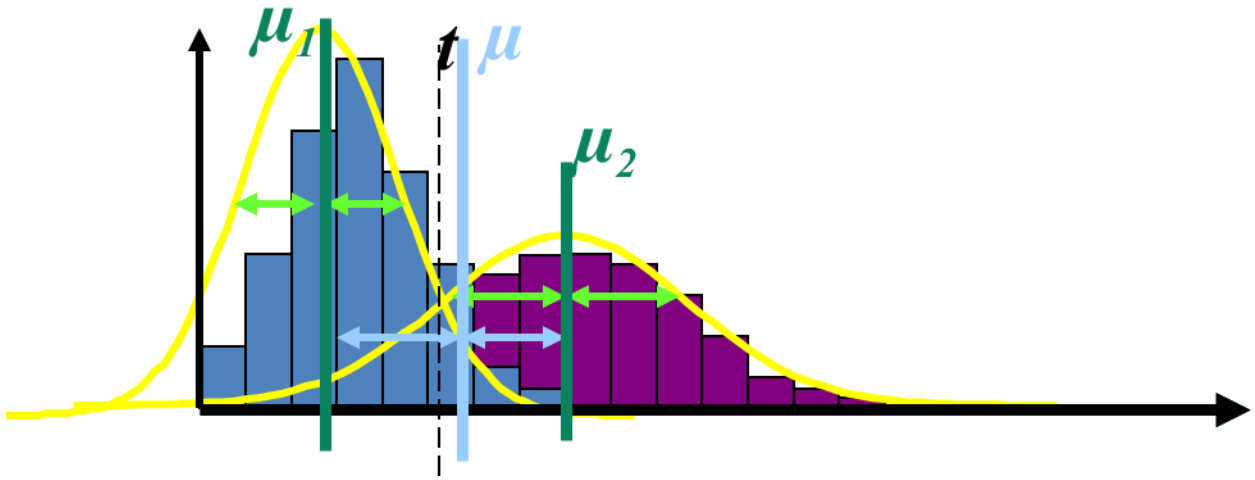
```
procedure histogram(I,H);
{
"Initialize the bins of the histogram to zero."
for i := 0 to MaxVal
   H[i] := 0;
"Compute values by accumulation."
for L := 0 to MaxRow
   for P := 0 to MaxCol
      {
      grayval := I[r,c];
      H[grayval] := H[grayval] + 1;
      };
}
```

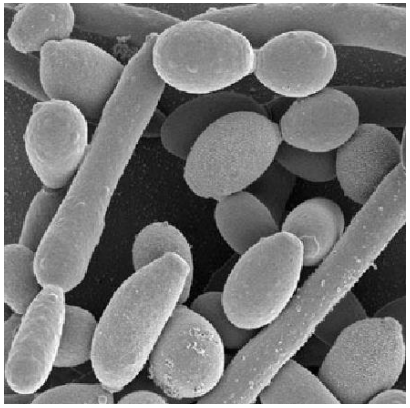
Binarização por Limiar Global

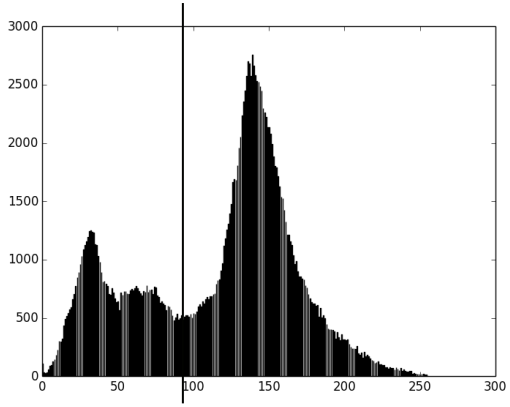


$$g(x,y) = \begin{cases} 1 & se \quad f(x,y) > T \\ 0 & se \quad f(x,y) \leq T \end{cases}$$

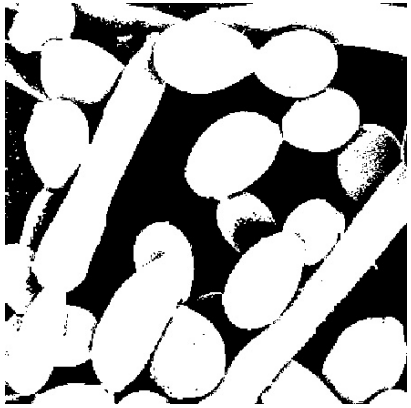
- Método de Otsu







T=99



- Método de Otsu
  - ***Ideia:*** minimização da variância intra-classes $\sigma_W^2 = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$

Probabilidade do grupo de pixéis com níveis $\leq t$

$$q_1(t) = \sum_{i=0}^{t} P(i)$$

$$q_2(t) = \sum_{i=t+1}^{MaxVal} P(i)$$

Probabilidade do grupo de pixéis com níveis $> t$

$$\sigma_1^2(t) = \sum_{i=0}^{t} \left(i - \mu_1(t)\right)^2 P(i) / q_1(t)$$

$$\sigma_2^2(t) = \sum_{i=t+1}^{MaxVal} \left(i - \mu_2(t)\right)^2 P(i) / q_2(t)$$

$$\mu_1(t) = \sum_{i=0}^{t} iP(i) / q_1(t)$$

$$\mu_2(t) = \sum_{i=t+1}^{MaxVal} iP(i) / q_2(t)$$

Frequência do nível de cinzento (i) na imagem

$$P(i) = h(i) / |R \times C|$$

  - Nota: equivalente à maximização da variância inter-classes

$$\sigma_B^2 = q_1(t)\left(1 - q_1(t)\right)\left(\mu_1(t) - \mu_2(t)\right)^2$$

$$\sigma^2 = \sigma_W^2 + \sigma_B^2$$

A equivalência resulta de a soma da variância intra-classes com a variância inter-classes é uma constante.

Cálculo Automático do Limiar - Algoritmo de Otsu

- Algoritmo de Otsu para determinação do limiar *t*
  - Iniciar

$$P(i) = h(i)/|R \times C| \qquad q_1(0) = P(0)$$

$$\mu = \sum_{i=0}^{MaxVal} iP(i) \qquad \mu_1(0) = 0$$

  - For *t := 0* to *MaxVal-1*

$$q_1(t+1) = q_1(t) + P(t+1)$$

$$\mu_1(t+1) = \frac{q_1(t)\mu_1(t) + (t+1)P(t+1)}{q_1(t+1)}$$

$$\mu_2(t+1) = \frac{\mu - q_1(t+1)\mu_1(t+1)}{1 - q_1(t+1)}$$

$$\sigma_B^2(t) = q_1(t)\big(1 - q_1(t)\big)\big(\mu_1(t) - \mu_2(t)\big)^2$$

  - Determinação do limiar

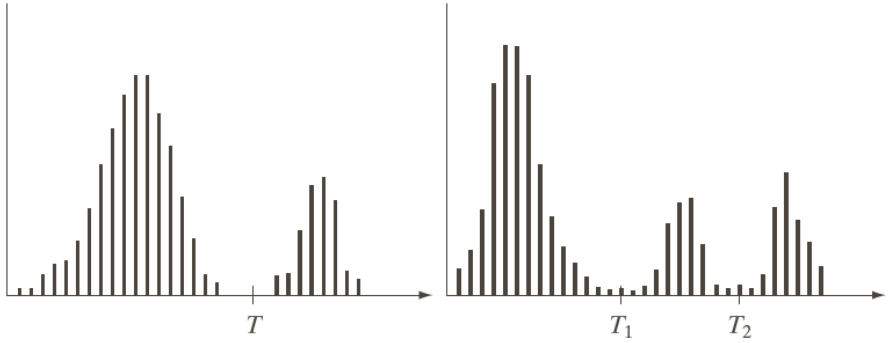
$$\hat{t} = \arg\max_t \sigma_B^2(t)$$

Original (*MaxVal*=255)

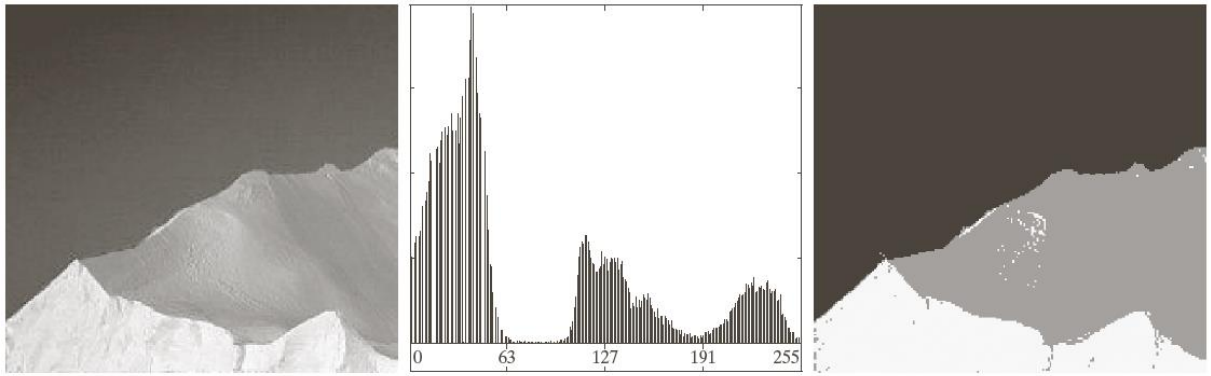
t = 93

Multiple Thresholding

https://scikit-image.org/docs/stable/auto_examples/segmentation/plot_multiotsu.html



$$g(x,y) = \begin{cases} a & se \quad f(x,y) \geq T_2 \\ b & se \quad T_1 \leq f(c,y) < T_2 \\ c & se \quad f(x,y) < T_1 \end{cases}$$



a b c

FIGURE 10.45 (a) Image of iceberg. (b) Histogram. (c) Image segmented into three regions using dual Otsu thresholds. (Original image courtesy of NOAA.)

- Técnicas de melhoramento
  - Filtragem
  - Utilização dos contornos

- Outras técnicas
  - Limiar variável (partição da imagem)
  - Limiar local baseado em vizinhos